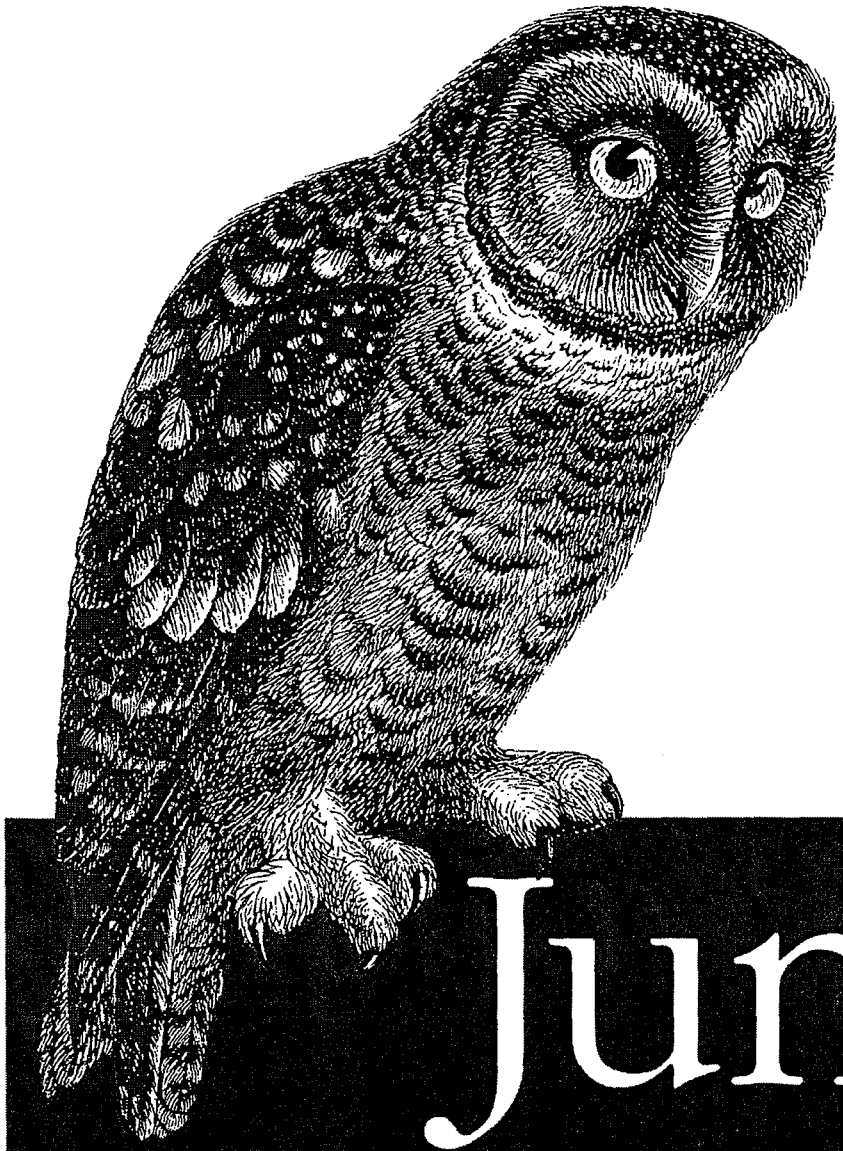
EXHIBIT A

A Practical Guide to Junos Routing and Certification

2nd Edition
Revised & Updated



Junos® Enterprise Routing

O'REILLY®

JUNIPER NETWORKS

Peter Southwick, Doug Marschke & Harry Reynolds

74.116.12.5
PBR public interface

Beer-Co has suppliers that use the Internet for connectivity, and one supplier, Oats.com, uses an IPSec-based VPN to interact with Beer-Co.

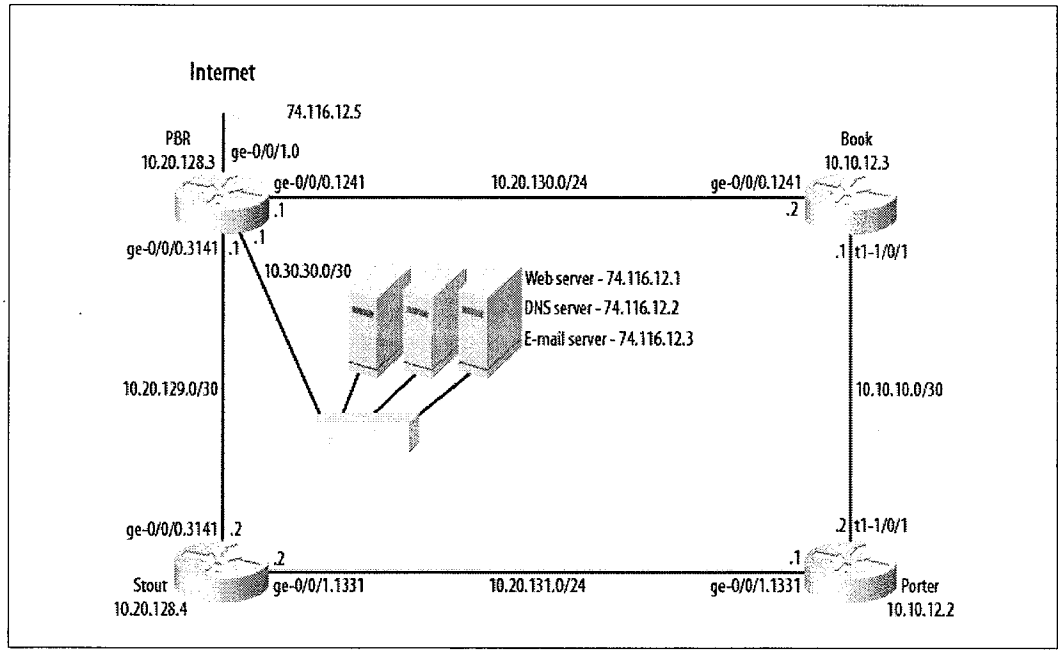


Figure 12-1. Internet access for Beer-Co Inc.

Packet- Versus Flow-Based Processing

Historically, Juniper Networks routers use a packet-based forwarding model, in which each packet is individually processed and routed. In contrast, the Juniper security devices are based on a flow model. Handling traffic as flows offers significant benefits for stateful services. In the flow model, the initial packets of a communication are subjected to various levels of packet security inspections and validity checks, in addition to a *single* route lookup. Once the packet is deemed permissible, a corresponding session state is installed into the forwarding plane to facilitate expedited forwarding for subsequent packets belonging to the same flow. In effect, the first packets are deeply scrutinized, and the remaining packets of the same session follow a fast path through the processing.

A *flow* is a unidirectional sequence of packets. The matching flow in the return direction is grouped to form a session, which is therefore composed of two unidirectional flows. The sessions reflect the applications that transit the firewall.

JUNIPER01919660

Architecture Changes

The addition of stateful security to Junos represents some significant changes in control plane capabilities through the introduction of new service daemons and in packet forwarding behavior with the addition of flow-based processing. This section provides a high-level overview of these changes.

Adding flow-based forwarding

One of the primary changes in Junos is the addition of flow-based processing. This is implemented along with the existing packet-based processing capabilities, such as stateless firewall filters. The changes in Junos result in a combination of packet- and flow-based treatments, as shown in Figure 12-2.

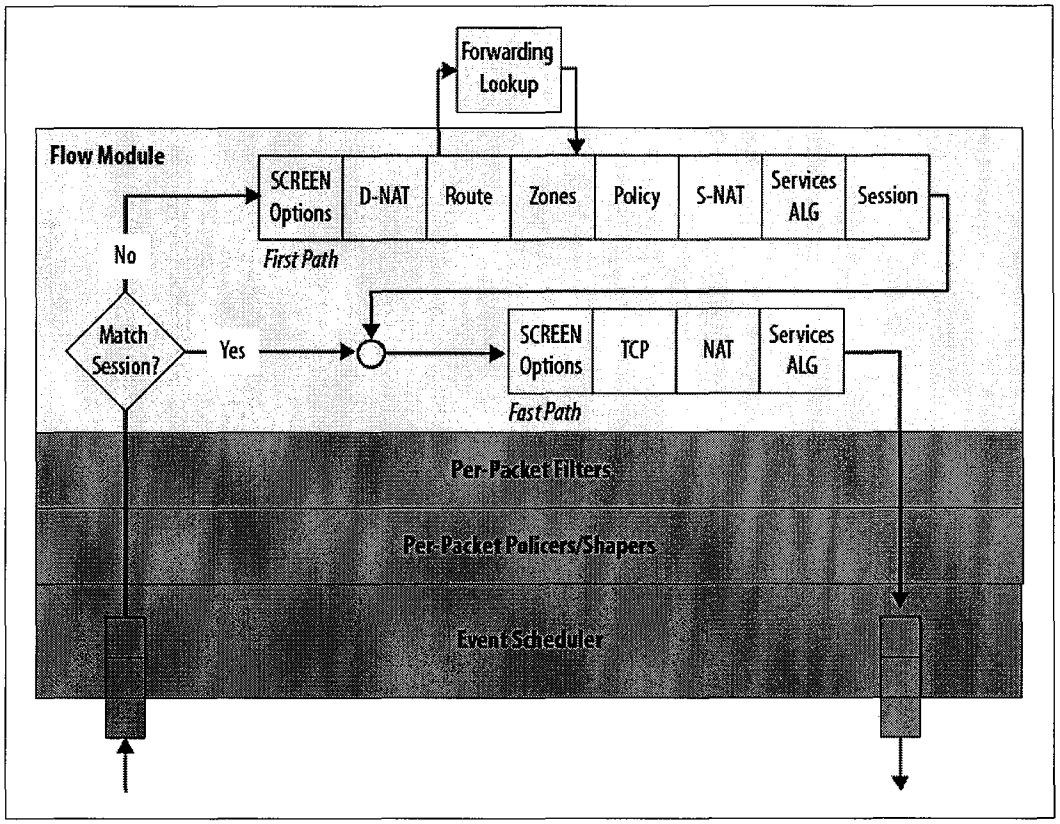


Figure 12-2. Combined packet- and flow-based processing

Figure 12-2 shows the packet and flow processing in Junos. An incoming packet is analyzed at the interface as a stateless entity for policers and firewall filters. If the packet passes these checks, the flow processing begins.

JUNIPER01919661

Flow-based processing is performed by the flow daemon (flwdd). This process runs in parallel with the other processes in Junos and uses extensive resources. If flow-based processing is disabled on the router, flwdd does not release the resources of the processor. Resource management is a concern for systems that perform resource-heavy tasks (e.g., full Internet feeds) and have flwdd. As of this writing, it is not possible to fully disable flwdd.

A flow is a unidirectional stream of related packets that meet the same matching criteria and share the same characteristics. Two flows are combined (ingress and egress) to form a session. Junos treats packets belonging to the same session in the same manner. Specifically, configuration settings that determine the fate of a packet—such as the security policy that applies to it, whether the packet is sent through an IPSec tunnel, or whether NAT is applied—are assessed for the first packet of a session. The resultant set of actions and services is applied to the rest of the packets in the session. The following criteria are used to determine whether a packet matches an existing session:

- Source address
- Destination address
- Source port
- Destination port
- Protocol

Flows and sessions. The stateful handling of traffic requires the creation of a session. A session is created based on the characteristics of the first packet in a flow. Sessions are used for:

- Storing security measures to be applied to the packets of the flow
- Caching information about the state of the flow—that is, logging and counting data for a flow is cached in its session
- Allocating required resources for features such as NAT and IPSec tunnels
- Providing a framework for features such as Application Layer Gateways (ALGs)

The combined effects of flow and session state bring together the following features and events that affect a packet as it undergoes flow-based processing:

- Flow-based forwarding
- Session management, including session aging and changes in routes, policy, and interfaces
- Management of VPNs, ALGs, and authentication
- Management of policies, NAT, zones, and screens

Each session resulting from a flow is associated with a timeout value. For example, the default timeout for the Transmission Control Protocol (TCP) is 30 minutes; the default

timeout for the User Datagram Protocol (UDP) is 1 minute. When a flow is terminated, it is marked as invalid, and its timeout is reduced to 10 seconds. You can change the idle timeout value; it is designed to ensure that system resources are not tied up indefinitely on an otherwise defunct flow.

Session timeouts are associated with specific services within Junos. Changing a timeout for a predefined service can cause unexpected results and should be done with caution. For some services (e.g., terminal emulation), the session might be open on a desktop for hours without traffic. In these cases the service timeout can be extended to accommodate this traffic pattern.

Junos security packet walk

In this section, we will follow a packet as it traverses the Junos data plane, where it encounters a mix of packet- and flow-based handling steps. Figure 12-2 shows the steps described in the following text.

The steps shown for the first path represent the full set of checks and service instantiations that you can perform against the initial packets of a session. In contrast, the fast path represents the streamlined steps executed for previously processed (and accepted) sessions. The two-stage approach provides the ability to deeply inspect initial packets, which is computationally expensive but needed for true security, while at the same time offering high throughput by switching permitted traffic based on established session state. It should be noted that not all packets need to be touched at all possible processing points. For example, NAT is optional, and when not configured, NAT processing is not evoked. The packet processing steps are as follows:

1. Accept an incoming packet, perform class of service (CoS) behavior aggregate (BA) classification, and note the ingress interface's zone for later policy lookup.
2. Process the packet through the ingress policer/shaper.
3. Evoke the multifield CoS classification or the firewall filter.
4. Perform a lookup session; if no match, follow the first path:
    a. Conduct a firewall screen check.
    b. Perform destination NAT as required for the incoming packet.
    c. Perform a route lookup to determine the egress interface.
    d. Locate the destination (outgoing) zone, based on the route lookup result.
    e. Look up and execute policy based on incoming and outgoing zones; results include permit, deny, and reject.
    f. Allocate the source NAT address to the packet.
    g. Set up ALGs as needed to support identified applications.
    h. Install a session tuple for fast path processing of related packets.

JUNIPER01919663

If a session is matched, follow the fast path:

a. Monitor the traffic for screen violations.
b. Perform TCP checks to look for connection anomalies and match responses.
c. Conduct NAT translation as required.
d. Perform ALG processing as needed.

5. Whether first or fast path, perform forwarding services on the packet based on the session information.
6. Perform egress firewall filtering, which can evoke a policer action.
7. Perform egress shaping or interface-level policing; schedule and transmit the packet.

Junos Security Summary

Integrating security features into Junos software is a significant milestone in the software's evolution. Looking back at Figure 12-2, you can appreciate the combined one-two punch of these services in Junos. You can now have the best of all worlds: the familiar Junos software CLI, its proven modular design that separates the control and data planes, the two-stage commit process, commit and operational scripts, and world-class routing protocol implementations. On top of this, you also get significant security and service features and enhancements.

The combined packet- and flow-based processing means that packet-based features relating to firewall filters, policers, and shapers, packet classification, queuing, and CoS continue to operate as before. Likewise, ASP-based platforms such as the M10i and M7i will continue to use the service configurations and modes described in Chapter 9 and Appendix A, which cover Layer 2 and Layer 3 services, respectively.

For users initially deploying devices with these security features, the reverse stance on denying versus accepting packet flows by default might take a bit of getting used to. The choice of router versus secure operating contexts helps to mitigate this issue and allows you to deploy *your* router so that it operates like a traditional router or as an integrated firewall router, as required by the needs of your network.

Understanding Junos Operational Modes

A J-series Services Router or an SRX Services Gateway can operate as either a stateful firewall or a router, depending on whether it is in the secure or router context:

Secure context
: This mode allows the device to act as a prudent stateful firewall. To allow traffic to pass through the device, you must explicitly configure a security policy for that purpose. In secure context, the router forwards packets only if a security policy

JUNIPER01919664

permits it. All transit traffic is processed as traffic flows and assigned to sessions when permitted by the policies.

All J-series routers and SRX Services Gateways are shipped from the factory in a secure context.

Router context
: This mode allows a router to act as a packet-based stateless router in which all management and transit traffic is allowed. In router context, traffic is handled in a per-packet mode of operation and no security policies are needed to provide connectivity.

Switching between secure and router contexts

Switching between secure and router contexts is performed by adding the packet mode commands to the security stanza. Once these commands are entered, all traffic is processed in a stateless manner. The remainder of the security stanza is effectively ignored. The commands are:

```
peter@pbr# show security
security {
    ...

    forwarding-options {
        family {
            inet {
                mode packet-based;
            }
            inet6 {
                mode packet-based;
            }
            mpls {
                mode packet-based;
            }
        }
    }
}
```

Default configurations

The default configuration on the J-series and SRX Services Gateways is model-dependent. Each SRX model has a different default configuration, as do the J-series routers. Our lab J-2320s have the following default configuration:

- The built-in Gigabit Ethernet interface, `ge-0/0/0`, is bound to a preconfigured zone called *trust*. All other interfaces are not bound to any zone.
- The `ge-0/0/0` interface is configured to allow management access with Secure Shell (SSH) and Hypertext Transfer Protocol (HTTP) services enabled. The following host-inbound services are configured for the `ge-0/0/0` interface in the trust zone:

JUNIPER01919665

  - —HTTP
  - —HTTPS
  - —SSH
  - —Telnet
  - —Dynamic Host Configuration Protocol (DHCP)
- TCP reset is enabled in the trust zone, and the default policy for the trust zone allows transmission of traffic from the trust zone to the untrust zone.
- All traffic within the trust zone is allowed.
- The following screens are enabled for the untrust zone:
  - —Internet Control Message Protocol (ICMP) Ping of Death
  - —IP source route options
  - —IP Teardrop
  - —TCP Land attack
  - —TCP SYN flood
- The default policy for the untrust zone is to deny all traffic.

The following commands load the factory default settings, which place the router into a secure context. There is no root password in the default configuration, so you must assign one using the `set system root-authentication` command before you can commit:

```
[edit]
peter@pbr# load factory-default
warning: activating factory configuration

[edit]
peter@pbr# show | no-more
## Last changed: 2010-11-28 15:44:55 PST
system {
    autoinstallation {
        delete-upon-commit; ## Deletes [system autoinstallation] upon change/commit
        traceoptions {
            level verbose;
            flag {
                all;
            }
        }
    }
    services {
        ssh;
        web-management {
            http {
                interface ge-0/0/0.0;
            }
        }
    }
    syslog {
```


JUNIPER01919666

```
        user * {
            any emergency;
        }
        file messages {
            any any;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
    ## Warning: missing mandatory statement(s): 'root-authentication'
}
interfaces {
    ge-0/0/0 {
        unit 0;
    }
}
security {
    screen {
        ids-option untrust-screen {
            icmp {
                ping-death;
            }
            ip {
                source-route-option;
                tear-drop;
            }
            tcp {
                syn-flood {
                    alarm-threshold 1024;
                    attack-threshold 200;
                    source-threshold 1024;
                    destination-threshold 2048;
                    queue-size 2000;
                    timeout 20;
                }
                land;
            }
        }
    }
    zones {
        security-zone trust {
            tcp-rst;
            interfaces {
                ge-0/0/0.0 {
                    host-inbound-traffic {
                        system-services {
                            http;
                            https;
                            ssh;
                            telnet;
                            dhcp;
                        }
                    }
```

```
                }
            }
        }
        security-zone untrust {
            screen untrust-screen;
        }
    }
    policies {
        from-zone trust to-zone trust {
            policy default-permit {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone trust to-zone untrust {
            policy default-permit {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    permit;
                }
            }
        }
        from-zone untrust to-zone trust {
            policy default-deny {
                match {
                    source-address any;
                    destination-address any;
                    application any;
                }
                then {
                    deny;
                }
            }
        }
    }
}
```

The default configuration is the starting point for introducing all the other features that are used to secure our router. In the following sections, router PBR is set to act as the Internet gateway for Beer-Co. Although not shown here, the initial configuration of the interfaces and routing protocols is performed as described in the previous chapters of this book. We are going to focus only on the security features.

JUNIPER01919668

Operational modes summary

The operating system in the J-series routers and SRX Services Gateways supports security features that were previously found only in purpose-built firewalls. These features allow the routers to operate as a prudent firewall in the enterprise. The same device can be converted to a stateless packet mode device with the introduction of a couple of commands. This allows these devices to be used in many different roles in the enterprise.

Security Features

In the following sections we explore the common security features that are associated with Junos. These features allow us to secure a network from Internet threats while providing connectivity for users of the enterprise. This balancing act is supported by security policies that permit or deny traffic through the gateway, network address translations that hide the internal structure of our network from prying eyes, virtual private network tunnels that encrypt traffic for transmission over the Internet, and threat detection schemes that block traffic that makes it through the initial lines of defense. We present only one possible scenario for securing a network; many other possibilities are being used in enterprises today.

Branch Office and Data Center SRXs

The full set of security features supported in Junos is found in the devices referred to as Branch Office SRX series services gateways (SRX240, SRX650, etc.). A subset of these features can be found in the J-series routers and a further subset found in the larger Data Center SRXs (e.g., SRX1400, SRX3Ks, SRX5Ks). The full definition of what models support which features can be found on the Juniper website.

The features presented in the following sections are found in the Branch Office SRXs and the J-series routers. For a treatment of the features that are found exclusively in the high-end SRXs, refer to *Junos Security* (O'Reilly).

Common feature set

The features that Beer-Co is using to protect their enterprise from the threats found in the Internet include a combination of stateful security polices, NAT, VPNs, and threat detection. We add these features to the existing configurations that have been built into the enterprise. Stateless firewall filters and interface policers are a part of the security plan for Beer-Co, but these have been covered previously and will not be repeated here.

Security policies

Stateful security policies are Beer-Co's second line of defense (the firewall filters that trap all obvious malicious traffic are the first line of defense). The stateful policies are

JUNIPER01919669

EXHIBIT B

MILLER & COMPANY REPORTERS

CERTIFIED COPY

UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.,	)	
	)	
Plaintiff,	)	
	)	
vs.	)	No. C 10-4234 SI
	)	
JUNIPER NETWORKS, INC.,	)	
	)	
Defendant.	)	
	)	

HIGHLY CONFIDENTIAL

30(B)(6) DEPOSITION OF: KRISHNA NARAYANASWAMY

TAKEN ON: September 20, 2011

NO. 12950 REPORTED BY: BRENDA L. MARSHALL
CSR No. 6939

with all of those products?

A. Yes. I'm familiar with all of those products.

Q. And you're familiar with the basic JUNOS architecture?

10:04:52

A. JUNOS is --

MR. MCPHIE: Objection. Vague and ambiguous.

THE WITNESS: Again, JUNOS is a very complex operating system --

10:04:56

BY MR. HOSIE:

Q. It is.

A. -- to the extent -- I'm -- I'm aware of the high-level architecture. If -- I'm not an expert on every line of code that is there in JUNOS.

10:05:06

Q. How big is the code base for JUNOS for the 11.3 release?

A. I do not know. It's in the millions of lines of code.

10:05:14

Q. Something like 195,000 files?

A. To be honest with you, I've not even gone and looked at that.

Q. All right, sir. You have actually, personally, during your tenure at Juniper,

05:24

configuration is what you meant by this is all configured in advance?

A. That's correct.

Q. Because the admin has to say, "Okay, here's what I want to do with this kind of traffic or that kind of traffic"?

10:19:46

A. Right. So I believe you generalized that statement. So what I was referring to was mostly around flow classification, how do you identify -- identify a flow.

10:19:58

Q. Okay. Which is one example of configuration, but there are other types of configuration, too?

A. There are other type of configuration, too.

10:20:04

Q. Sure.

A. And right now, we are looking at step one, which is classifying a packet, the first packet that comes in of a new flow.

Q. Okay. So a first packet comes in, and the box will have been configured for classification as per the admin's choices?

10:20:10

A. The box would be configured for classification per the admin's choices. That's correct.

:20:24

Q. Okay. So given that, what happens when the first packet comes in? And let's assume it's a new flow, not a subsequent packet of an already processing flow. You understand?

10:20:36 MR. MCPHIE: Objection. Vague and ambiguous.

BY MR. HOSIE:

Q. First packet of a new flow, what happens?

10:20:40 MR. MCPHIE: Same objection. Incomplete hypothetical.

THE WITNESS: So when the first packet of -- of a new flow comes in, there is -- there's a policy lookup that happens, and the 10:20:58 policy lookup -- depending on, again, what the configuration of the box is, the policy lookup will determine what actions need to be taken so that a predefined set of actions that can be taken, which are configured, again, by the 10:21:12 administrator on this session.

BY MR. HOSIE:

Q. Give us examples of actions that can be configured.

A. Again, this is an ambiguous question. :21:24 Can you be more specific as to are we still in

change edited -- edit -- an edit phase of the policy, and once policies have been edited, they have to be committed. The commit is what makes the change.

10:33:06 Q. Okay.

A. And once policy changes are committed, any new policy -- policy lookup is done on the first packet. As I said, the first packet that identifies the policy is based on this

10:33:20 preconfigured set of actions that need to be taken. So for any new policy lookups, you would now take the changed policy.

Q. Okay. So on a next new flow basis, though; right?

10:33:34 MR. MCPHIE: Objection. Vague and ambiguous.

BY MR. HOSIE:

Q. Do you understand my question?

A. Can you complete that question?

10:33:38 Q. Sure. So let's assume the admin makes a policy change, pushes the commit button, policy change is made, will the next new flow that's germane to that policy be treated differently, or will it affect a flow already in process?

10:33:56 MR. MCPHIE: Objection. Compound.

Incomplete hypothetical.

THE WITNESS: So there were multiple questions there so I'll answer the first one.

BY MR. HOSIE:

10:34:02 Q. Please.

A. The first -- so any new flow that is -- that requires the policy lookup would take the effect of the new policy.

Q. Okay. What about a flow that's already

10:34:12 in process when the admin hits the commit, what happens with the in-process flow? Does the processing change or not?

MR. MCPHIE: Objection. Compound.

BY MR. HOSIE:

10:34:26 Q. If you know.

A. I do not recall.

Q. Okay. So going back to our document, the J-series router, I see a reference here to a flow state.

10:34:56 A. Can you tell me which page?

Q. Sure. It's subparagraph 2.

A. Yes. I see that.

Q. "The first packet in the flow is used to create the flow state." What's a flow state, please?

10:35:08

policy lookup?

A. Each service has a separate policy lookup.

Q. Okay. So the first packet comes in, there's a policy lookup, and that lets the system know what the policies are for the packet at issue?

MR. MCPHIE: Objection. Vague and ambiguous. Compound. Asked and answered.

THE WITNESS: Yes. When the first packet comes in, the -- the policy lookup happens, and depending on the services that are enabled, each service has its own policy.

BY MR. HOSIE:

Q. Okay. And a service might be, for instance, a firewall functionality?

A. A service, for example, can be firewall functionality. That's correct.

Q. Okay. And then once this policy lookup takes place, what's the next step?

MR. MCPHIE: Objection. Vague and ambiguous. Incomplete hypothetical.

THE WITNESS: Once the policy lookup is completed, the set of actions associated with the policy is enforced on the packet and, as the

paragraph says, that information is kept for multiple packets, the subsequent packets of that flow.

BY MR. HOSIE:

10:37:36 Q. So flow-based stateful processing, what does the word "stateful" mean to you here?

A. Stateful -- what -- what it -- what "stateful" means to me in this context is keeping track of that predefined set of actions

10:37:52 that an operator of an admin has configured in the policy to -- to be enforced on subsequent packets of that flow.

Q. So a first packet comes in, there's the policy lookup part of the classification

10:38:08 process, and then the set of actions is enforced on the packet, and then state is maintained so that subsequent packets of the same message are treated the same way --

MR. MCPHIE: Objection.

BY MR. HOSIE:

Q. -- correct?

MR. MCPHIE: Objection. Compound. Asked -- asked and answered.

THE WITNESS: I think --

10:38:26 MR. MCPHIE: Vague and ambiguous.

BY MR. HOSIE:

Q. Okay. And how does the system then know what to do with other packets that are part of the same message?

10:39:24 MR. MCPHIE: Objection. Calls for a legal conclusion. Vague and ambiguous.

THE WITNESS: Can you please clarify what you mean by "message." First time I see the word "message" pop up today.

10:39:36 BY MR. HOSIE:

Q. Okay. Would you be more comfortable with the word "flow"? What -- let me ask the question. How does the system know what to do with subsequent packets of the same flow?

10:39:44 MR. MCPHIE: Objection. Vague and ambiguous.

THE WITNESS: So based on the first packet, the policy lookup on the first packet, and the set of predefined actions that are

10:39:56 associated with the first packet, that set of actions is maintained in the flow state for subsequent packets of that flow.

BY MR. HOSIE:

Q. That set of actions is maintained in the

10:40:08 flow state; right?

MR. MCPHIE: Objection.
BY MR. HOSIE:
Q. That's what you just said?
MR. MCPHIE: Objection. Vague and ambiguous.
10:40:14
THE WITNESS: That's at a -- predefined actions that were configured by the policy.
BY MR. HOSIE:
Q. All right.
10:40:26
A. Start with the first predefined set.
Q. Okay. Did your lawyers tell you to use the word "predefined" as many times as you could, sir?
MR. MCPHIE: I'll instruct you not to
10:40:34
answer that.
THE WITNESS: Not as --
MR. HOSIE: I'll strike the question.
Q. So a set of actions maintained in the flow state, mechanically, tell me exactly how
10:40:42
that works. How does -- how does the system maintain the set of actions in the flow state?
MR. MCPHIE: Objection. Compound. Vague and ambiguous.
THE WITNESS: So, again, there's a set
10:40:56
of predefined actions associated with --

BY MR. HOSIE:

Q. Okay.

A. So maybe we should be on the same terms here.

10:45:12 Q. Module is a C file, and it's a piece of code that does something particular.

A. Okay. So in this particular example, yeah, that is a C file that does permit/deny and code packets.

10:45:26 Q. So -- and if the admin configures a system to make that relevant permit/deny and byte processing, the system would then use that module?

A. No. The system -- so to answer your

10:45:38 question, the system always uses a module. It's not like only when it's configured it uses the module. It just bypasses certain things if -- based on the predefined rules.

Q. Got it. So the modules -- I'm going to

10:45:50 explore that with you, but as I understand it, the modules -- all modules are always present, but how and when they're used is a function of the policies?

A. The --

10:45:58 MR. MCPHIE: Objection.

THE WITNESS: The predefined policies.

MR. MCPHIE: Objection. Vague and ambiguous.

BY MR. HOSIE:

10:46:02 Q. The predefined policies; correct?

MR. MCPHIE: Objection. Vague and ambiguous. Compound. Calls for a legal conclusion.

THE WITNESS: So when we ship the

10:46:14 product, the product has the entirety of all the actions that can be taken, and based on the predefined policies for a given flow, certain actions are taken or not taken.

BY MR. HOSIE:

10:46:32 Q. Okay. And within the innards of the system, what does that mean, certain modules are used -- or I think your word was "bypassed"; correct?

MR. MCPHIE: Objection. Vague and ambiguous.

THE WITNESS: So, again, as I said, when -- what I mean by "module" is it's a C file, and C file is not a component. C file -- C files are created for division of labor.

:46:56 And so, again, I'm -- I'm not able to

answer your question because I don't agree -- I feel we don't agree on the term "module."

BY MR. HOSIE:

Q. Okay. Let me -- let me ask it differently, then. So all the actions are -- are always loaded with the product? They all come with; right?

10:47:04

MR. MCPHIE: Objection. Compound. Vague and ambiguous.

THE WITNESS: Again, I don't understand the word "loaded." There is an executable image that has binary code that can execute all the actions.

10:47:16

BY MR. HOSIE:

Q. Okay. And then --

10:47:26

MR. MCPHIE: By the way, we've been going about an hour so --

MR. HOSIE: Let me just finish this line.

MR. MCPHIE: -- whenever you're ready for a break, that would be great.

10:47:30

BY MR. HOSIE:

Q. And which actions, then, are implemented and which not are a function of the admin config, as reflected in the policy?

10:47:38

Mischaracterizes prior testimony.

THE WITNESS: That's one way of -- of looking at functions performed in the --

BY MR. HOSIE:

11:04:40 Q. And --

A. -- in the system.

Q. Thank you. And the configuration of the system drives which actions are performed versus not performed?

11:04:48 MR. MCPHIE: Objection. Vague and ambiguous.

THE WITNESS: The configuration of the policies in the system determine what actions are performed and what are not.

11:05:00 BY MR. HOSIE:

Q. All right, sir. Now, for any of the Juniper products, are actions just undertaken randomly, haphazardly, chaotically?

A. I think that's an ambiguous question. I 11:05:16 don't know -- can you be more specific as to what you mean by "randomly"? You mentioned three different words there.

Q. Are --

A. What -- what is exactly the thing that :05:22 you're looking for?

answer this because you threw the word "process" in. I don't know what you mean by "process." So I can state what happens.

Q. Please.

11:15:48 A. When subsequent packets come in, the flow state that -- that is stored in memory is -- is looked up, and the same set of actions, the predefined actions, that were performed on the first packet is also performed on the

11:16:04 subsequent packets of that flow.

Q. So you don't have to go through the same classification and policy lookup for every packet; you only do it for the first packet, and the actions are stored in memory?

11:16:16 MR. MCPHIE: Objection. It's compound. Vague and ambiguous.

THE WITNESS: The result of the first packet, the policy lookup and the actions that need to be taken, based on -- on the

11:16:30 configuration that was set up by the admin is -- is stored in the flow state, and so the subsequent packets then are -- avoid the policy lookup and used that flow state to have the same set of predefined actions enforced on them.

1:16:48 BY MR. HOSIE:

a new flow has a policy lookup that is done, and the policy lookup identifies a list of predefined actions, which are then stored in the flow state for subsequent packets of that flow.

11:19:58 BY MR. HOSIE:

Q. How are they stored? How are the actions stored in the flow state?

A. The actions are stored as data structures in memory.

11:20:04 Q. The actions are stored -- what do you mean by "the actions are stored as data structures in memory"?

A. Depending on the algorithm that is being used to enforce that particular action, there is

11:20:18 a -- there is a data structure, I don't know how it's fixated, in memory, with some encodings that say --

Q. Do this?

A. -- this value means do this, this value

11:20:32 means do that.

Q. Okay. And so those -- okay.

And then so in the next packet of the -- of the current flow comes in, it just runs through those actions?

:20:44 A. In fact, the first packet also runs

through that.

Q. Of course. Every packet of the flow does?

A. Every packet. Exactly.

11:20:48 Q. All right. So is -- for the J-series routers, are these session-based?

A. So the J-series routers can run in two modes. There's a packet-based mode, as well as a flow-based mode.

11:21:04 Q. Okay. Okay. And we've been talking about the flow-based mode?

A. So in the context of flow processing, we have been talking about the flow-based mode.

Q. And a flow-based mode is where there is

11:21:14 a classification process deriven by the first packet, and all subsequent packets of the same flow are treated the same way. That's flow-based, in your lexicon?

MR. MCPHIE: Objection. Compound.

11:21:24 THE WITNESS: Flow-based, in my lexicon, is identifying of a flow based on parameters that are either in the packet or derived from the packet.

BY MR. HOSIE:

:21:32 Q. The first packet?

specifically for the J-series routers.

Q. Okay.

A. And in that context, yes, the Voyager project was the project that enabled the flow-based services.

11:26:50

Q. And it did that by actually changing the JUNOS operating system? That's how you made it happen; right?

A. So it was new functionality added to the JUNOS operating system.

11:27:02

Q. Thank you. Okay. Now, when you started working on the Voyager product, were there any Juniper products on the market that offered flow-based routing?

A. So I do not recall any Juniper products, but flow-based forwarding has been there since the early to mid '90s. In fact, my startup company, which was founded in 1997, did flow-based forwarding and routing.

11:27:20

11:27:34

Q. Known as disserve?

A. No. Disserve is totally different.

Q. Okay.

A. Disserve has something to do with quality of service.

But -- but flow-based routing, as I

:27:44

subsequent packets -- for that packet, as well as the subsequent packets of the flow to get the same treatment --

BY MR. HOSIE:

11:36:08 Q. So you don't have to --

A. -- as defined.

Q. -- go through -- yes. Thank you. So you don't have to go through this lookup process packet by packet by packet?

11:36:14 MR. MCPHIE: Objection. Asked and answered. Vague and ambiguous.

THE WITNESS: The flow state is stored in memory so that the policy lookup need not happen on a packet-by-packet basis for a given

11:36:26 flow.

BY MR. HOSIE:

Q. Thank you. And that's the efficient way of doing it; right? In a flow-based model?

MR. MCPHIE: Objection. Vague and

11:36:34 ambiguous.

THE WITNESS: That's one way of doing it.

BY MR. HOSIE:

Q. And that's how Juniper does it?

:36:38 MR. MCPHIE: Objection. Vague and

A. Machine to machine.

Q. Okay. So let's assume I'm on a Windows machine inside a Juniper-enabled network, all right, and I am looking at the Juniper FTP site, but I'm also downloading some e-mail. How does your system keep e-mail packets distinct from the FTP site packets?

MR. MCPHIE: Objection. Vague and ambiguous.

THE WITNESS: So at the bottom, the system deals with flows and sessions. And a session, as is specified on this document, consists of certain information that are there in the packets, like the IP addresses, the port numbers. Minimally, we have this notion of five tuple. I'm not sure if you have ever heard of this term, "five tuple" --

BY MR. HOSIE:

Q. Yes.

A. -- which stands for the source IP address, destination IP address, protocol, source port, and dstport.

Q. Yes.

A. So in the example that you gave of an e-mail versus FTP, the five tuples are

distinct --

Q. Yes.

A. -- and that's how a system like the J-series router keeps track of them.

11:43:28 Q. So the e-mail would have one set of five tuples, and my FTP flow would have a different set of five tuples?

A. The FTP session would have its own set of five tuples, and the email session would have

11:43:40 its own set of five tuples.

Q. So it's unique, session by session?

MR. MCPHIE: Objection. Vague and ambiguous.

MR. HOSIE: Let -- let me rephrase that.

11:43:44 Q. The system, being session-based, keeps sessions separate because each session is distinct and unique versus other sessions?

MR. MCPHIE: Objection. Compound.

THE WITNESS: Generally, your statement

11:43:56 is right, but I wouldn't say it's completely accurate. There are cases where there is a relationship between sessions. So in that sense, it's not unique.

BY MR. HOSIE:

.44:06 Q. Fair -- fair point. Fair point. Fair

classifies the packet, looks up the policy, looks up the -- the flow table, and if in the first packet, then it -- it identifies a policy.

Again, I want to go back to the term "policy." Because this is a distributed system, policy functions are also distributed. So there's no one policy lookup. There are multiple policy lookups. So one part of the policy lookup happens on the NPU.

13:19:30 Q. Okay. And you said "flow table." This morning, we were talking about "flow state." Are they synonyms, in your mind?

A. So when we talked this morning of a system that has multiple sessions of flows --

13:19:44 Q. Yes.

A. -- a collection of sessions of flows is called a flow table.

Q. Okay. Whereas the status of a particular flow is the flow state?

13:19:52 A. The flow -- the status of one particular flow is a flow state.

Q. Got it. Whereas the collection of all the flows running through the system is -- is monitored by what is called a flow table?

20:00 A. The status of all the -- the -- the

BY MR. HOSIE:

Q. Okay. All triggered by the first packet -- first packet classification, lookup and --

13:21:10 A. So the first packet classification looks up the flow table, and if there is no entry, then a policy lookup is done, which identifies the predefined set of actions, which happens to be a subset, in this case, because it's a 13:21:26 distributed system, they're applicable to this NPU, and that is cached in the flow table.

Q. As instantiated data structures?

MR. MCPHIE: Objection. Vague and ambiguous.

13:21:38 THE WITNESS: They're stored -- they're stored in memory as allocated data structures.

BY MR. HOSIE:

Q. Okay. And it's that storage and memory allocated data structure that gives the system 13:21:48 the ability to mainstay for flow-by-flow processing?

MR. MCPHIE: Objection. Vague and ambiguous.

THE WITNESS: The flow table in the :22:00 network processing unit, as the state required,

for subsequent packets of a given flow, to take the same predefined set of actions that were there in the first packet.

BY MR. HOSIE:

13:22:14 Q. Thank you. And by "predefined," you mean a set of actions chosen by a system admin?

MR. MCPHIE: Objection. Calls for a legal conclusion. Vague and ambiguous.

THE WITNESS: By "predefined," I mean

13:22:24 set of policy configurations that are made by an admin and that are in the box prior to the arrival of the packet.

BY MR. HOSIE:

Q. So the actions that the admin wants to

13:22:34 have versus the actions the admin does not want to have?

MR. MCPHIE: Objection. Compound. Asked and answered. Vague and ambiguous.

THE WITNESS: Let's put it this way. It

13:22:44 is the set of actions that the admin wants to have. You don't store what the admin does not want to have.

BY MR. HOSIE:

Q. Yeah. You pick what you want to happen;

13:22:52 you don't say, "Don't do this other stuff"?

algorithm can be used.

Q. Okay. So you're basically looking at the system dynamically and figuring out the best SPU to assign a particular flow to, given load constraints and other such factors?

13:25:04

MR. MCPHIE: Objection. Calls for a legal conclusion. Vague and ambiguous. Compound.

THE WITNESS: The central point's function is to pick an appropriate service-processing unit to anchor the session.

13:25:10

BY MR. HOSIE:

Q. Okay. And once that selection is done, what does the SPU do with the first packet?

A. So once that selection is done, the SPU does exactly the same processing that we discussed for the J-series router. So it does the -- for the first packet, it does the policy lookup and then identifies the predefined set of actions to be taken --

13:25:20

13:25:34

Q. Okay.

A. -- and stores that in the flow table. So it's, again, exactly what happens in the J-series.

Q. Okay. Got it. So, really, what's

.25:42

THE WITNESS: The service chain is not hard baked into the system. The order in which services are executed is baked into the system.

BY MR. HOSIE:

13:28:42 Q. Okay.

A. The service chain, again, is determined by the policy at run time.

Q. Good point. Because you don't know which services are implemented or not?

13:28:48 A. You may assume that. Yeah.

Q. Yeah. Thank you. I get it. Okay. But the order of the services is baked in pre run time?

MR. MCPHIE: Objection. Vague and

13:29:00 ambiguous.

THE WITNESS: So once a policy lookup tells you what services need to be run, the admin does not have control as to on what -- in what order they run.

13:29:10 BY MR. HOSIE:

Q. Okay. Okay. So let's -- if we assume services one through five, and let's say the admin says, "Okay. I'm going to implement one, three, and five," the system will know that it

:29:22 should go one then three then five?

A. If you --

MR. MCPHIE: Objection. Vague and ambiguous. Incomplete hypothetical.

THE WITNESS: Yeah. If you assume -- again, I believe you're inferring that one to five is ordered in that way. If you assume that one to five was ordered in an orderly fashion -- because somebody could order it any random way they want -- but if you assume that one to five is ordered, then if the policy says that the services that need to be run are one, three, and five, it will be run in that order, one, three, and five.

BY MR. HOSIE:

Q. Okay. So the first packet hits the SPU -- well, it's assigned to the SPU policy lookup, as dis -- as we discussed this morning, and then the actions chosen to be performed are instantiated as data structures in memory?

A. For the --

MR. MCPHIE: Objection.

BY MR. HOSIE:

Q. For the first and subsequent packets.

MR. MCPHIE: Objection. Compound. Vague and ambiguous.

THE WITNESS: When -- when a packet is received and a policy lookup is -- is performed and as a result of the policy decision you get the predefined set of actions, that is stored in the flow state as data structures in memory. That's correct.

BY MR. HOSIE:

Q. Thank you. Turn to the next page, please. There's a figure here, figure 3.2. Do you see that?

A. Yes, I do.

Q. Could you walk us through what is depicted here, please.

A. Okay. So starting from the left side, the packet -- bottom left side -- when a packet comes in, it comes into the I/O card.

Q. Okay.

A. And in the I/O card, the NPU functions that we just discussed happens. Okay. The -- the I/O card has an internal switch fabric, but -- okay. So let me back up.

This picture is an architecture that is evolved. So I got a bit confused. So some of the placement of the -- the modules are a little bit different.

was marked Plaintiff's Exhibit 4 for identification by the Reporter, a copy of which is attached hereto.)

MR. HOSIE: Oh, you may have two. That's great. Another copy. You have the one with the sticker?

THE WITNESS: I have the one with the sticker.

MR. HOSIE: That's good. That's the one we have to make sure our -- thank you. We -- we lawyers have our own protocols.

Q. Earlier today, I -- I had warned you that I was going to show you a list of Juniper products and ask you some questions about it.

A. Yes.

Q. I'm going to ask you first, which of these products use flow-based packet processing, as per our discussion about Viking IDP and the J service routers this morning?

MR. MCPHIE: Objection. Compound.

BY MR. HOSIE:

Q. Do you understand the question, sir?

A. Yes, I do. So this is specifically the Viking and the -- the J-series?

Q. Yeah.

A. The flow-based processing that we discussed?

Q. Yes. I want to know which of these products used flow-based processing.

14:30:04 MR. MCPHIE: Objection. Compound. Vague and ambiguous.

BY MR. HOSIE:

Q. And if it's easier for you to tell me which do not, do that as well.

14:30:12 A. Okay.

Q. I don't know what the default would be here.

A. The first page, none of them use.

And the second page, up to 53, none of

14:30:46 them use any flow-based processing.

Now, once you start the security category, the J-series -- as we talked about this morning, the J-series, as well as the three-digit SRXs, support both packet as well as

14:31:02 flow, but the -- going on to page No. 9 --

Q. Yes.

A. -- the NetScreen boxes are not JUNOS boxes, but they do deploy flow-based technology.

Q. Okay. MX?

..:31:22 A. The MX, that service is DPC. The

service is DPC. The multiservice is DPC, is the one card where the flow-based processing happens.

Q. And is that -- that's just flow or both flow and packet?

14:31:36

A. It can do both flow and packet.

Q. Okay.

A. And that is true of all M as well as T, all the way to 19.

Q. Okay. So they all are flow-based?

14:31:44

A. Right.

Q. Okay. Twenty?

A. Twenty on to 33 are not JUNOS-based, but they do deploy flow-based.

Q. Okay.

14:31:58

A. Thirty-four and 35, again, is not JUNOS, but does flow-based processing. And 36 through 45 are all flow-based.

Q. Okay.

A. And with the caveat that the three-digit SRXs do packet as well as flow.

14:32:14

Q. Okay. Whereas the four-digit just do flow?

A. Only do flow-based.

Q. Okay. So the first page, Application

14:32:22

I, BRENDA L. MARSHALL, Certified Shorthand Reporter, License No. 6939, do hereby certify:

That, prior to being examined, the witness named in the foregoing deposition, to wit, KRISHNA NARAYANASWAMY, was by me duly sworn to testify the truth, the whole truth and nothing but the truth:

That said transcript was taken down by me in shorthand at the time and place therein named and thereafter reduced to computerized transcription under my direction.

I further certify that I am not interested in the event of the action.

WITNESS this 30th day of September, 2011.

Brenda L. Marshall

BRENDA L. MARSHALL

EXHIBIT C

UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.,	)	
	)	
Plaintiff,	)	
	)	
vs.	)	No. C 10-4234 SI
	)	
JUNIPER NETWORKS, INC.,	)	
	)	
Defendant.	)	
	)	

HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY

DEPOSITION OF: OLIVER TAVAKOLI

TAKEN ON: June 19, 2012

13145

BRENDA L. MARSHALL

CSR No. 6939

10:29:00 Q. So let's -- let's, then, discuss
10:29:03 technology of, say, the SRX products. What's a
10:29:07 plug-in, in that context, sir?
10:29:08 A. A plug-in is a discernible module that
10:29:13 has some boundaries to it.
10:29:15 Q. Can you give me some examples of
10:29:18 plug-ins?
10:29:19 A. Protocol parser, deep packet inspector,
10:29:27 application identifier, application layer
10:29:33 gateway, ALGs.
10:29:37 Q. How many plug-ins come with the SRX
10:29:40 product?
10:29:43 MR. KAGAN: Objection. Vague.
10:29:45 THE WITNESS: [redacted]
10:29:47 [redacted]
10:29:51 [redacted]
10:29:55 [redacted]
10:29:57 [redacted]
BY MR. HOSIE:
10:29:59 Q. Okay. When you say "more the modules
10:30:02 that make up the system," what do you mean?
10:30:04 A. These are the building blocks out of
10:30:06 which the system is built.
10:30:07 Q. The plug-ins are the building blocks out
10:30:10 of which the system is built?

11:26:45 Q. So the first packet comes in. Is there
11:26:49 a step in the process where the system looks at
11:26:51 the policy, figures out what the processing will
11:26:53 be, and then allocates enough memory for that
11:26:56 particular processing for that particular flow?
11:26:59 MR. KAGAN: Objection. Vague and
11:27:00 compound.
11:27:03 THE WITNESS: I -- yeah. I -- I
11:27:04 wouldn't put it that way. I mean, the policies
11:27:07 themselves are all in memory. These systems
11:27:09 don't have disk on the -- on the SPCs. So
11:27:13 everything is in memory. Right? The policies
11:27:16 are all in memory. It's just a question of
11:27:18 finding which one, right, and having found that
11:27:21 policy, you don't need to make a copy of it for
11:27:23 this particular flow. If you had a million
11:27:25 flows that were going off of that same policy,
11:27:27 it would be a million flows going off of that
11:27:29 same policy.
11:27:30 BY MR. HOSIE:
11:27:30 Q. How -- let's assume there's a million
11:27:32 flows going off of the same policy. How were
11:27:34 each of those flows kept, distinctly?
11:27:36 A. There are -- there -- as I say, there's
11:27:38 a preallocation. It's kind of a static

11:28:30 flow?

11:28:31 MR. KAGAN: Objection.

11:28:32 THE WITNESS: It's not -- it's not dynamically allocated. It's static -- it's a slot assigned to that particular flow.

11:28:35 BY MR. HOSIE:

11:28:35 Q. Okay. Okay. And is that how, in our illustration, a million different flows are kept distinctly?

11:28:43 A. Yes.

11:28:43 Q. And how does that work? I mean, how are they kept distinctly?

11:28:47 MR. KAGAN: Objection. Vague.

11:28:48 THE WITNESS: The -- so -- I'm not -- again, I'm not quite sure what you're trying to get at. Each flow has a unique characteristic. Right? Even though you have a wild-carded rule, wild-carded rule says from this IP address to this IP address, but on any ports. Right? So the first flow that might arrive -- the first session -- TCP session that might arrive might be from port 1 to port 2.

11:29:12 BY MR. HOSIE:

11:29:12 Q. Yes.

11:29:13 A. The next one that might arrive might be

12:48:12 external corollaries, right --

12:48:15 Q. In terms of --

12:48:15 A. -- so you just -- in terms of you don't tend to think of, well, I'm going to enable protocol parsing, but protocol parsing is a plug-in that --

12:48:24 Q. Okay. Thank you. Let me -- let me be --

12:48:26 A. Yeah.

12:48:26 Q. -- a little more precise in my question.

12:48:28 In terms of what a system admin would see in configuring the system, could you give me a list of those plug-ins.

12:48:33 A. Again, I can give you a list of features.

12:48:36 Q. Okay.

12:48:36 A. You wish to only have a list of features that are implemented as plug-ins?

12:48:43 Q. Yes.

12:48:44 A. I believe IPS is, ALGs are, App ID, App Firewall, App QOS, I think, is the other one, App DOS.

12:48:54 Q. Okay. And a system admin can configure a box so that a number of these plug-ins are to be used; correct?

12:49:17

12:49:25

12:49:30

12:49:31 A. I think a system administrator
12:49:34 configures a box toward a particular end,
12:49:39 functional end, as we kind of have been talking
12:49:42 about use cases before, and as a result of that,
12:49:46 our developers, basically, having considered all
12:49:50 of the combinations of those features, decide
12:49:55 which plug-ins ought to be enabled as -- as a
12:49:58 result in -- in response to the configuration
12:50:02 combinations.
12:50:03 Q. Okay. When you say plug-ins enabled,
12:50:05 what do you mean?
12:50:06 A. I mean by -- by saying that plug-ins are
12:50:10 enabled, it basically means that based on
12:50:13 policy, based on the set of features you have
12:50:15 selected for a given five tuple selector that
12:50:22 you specified in the policy, you may have
12:50:26 enabled three features that may result in a very
12:50:31 particular sequence of components being enabled
12:50:39 and, in fact, being past the traffic of that
12:50:43 flow.
12:50:44 And, again, the point here is that --
12:50:46 that that's not limited to the plug-ins that are
12:50:50 kind of explicitly obvious. Right? If I enable
12:50:54 IPS, it's -- you would say, well, it's kind of
12:50:57 obvious that the IPS plug-in better get the

12:51:00 traffic, but there are a whole bunch of
12:51:02 ancillary plug-ins that have to do
12:51:05 preprocessing, postprocessing that are part and
12:51:07 parcel of -- of the set of plug-ins that are,
12:51:11 quote, enabled.
12:51:12 Q. Okay. I understand. And so if I'm the
12:51:16 system admin, if I say IPSec, that's going to
12:51:19 cause -- that's a feature, and it's going to
12:51:21 cause the machine to do something to make sure
12:51:23 that the necessary sequence of components will
12:51:25 be called for that traffic?
12:51:27 A. Yeah. I mean, with IPSec, it's --
12:51:30 it's -- it's a little more difficult. There are
12:51:32 things called route-based policies, there are
12:51:36 basically explicit policies that you -- that you
12:51:39 put in that are not route-based.
12:51:43 IPSec configuration, in general, is a
12:51:45 little bit more complex and requires a fair
12:51:47 amount of special sauce for the developer to
12:51:49 kind of translate that policy into exactly what
12:51:52 happens when it recovers.
12:51:55 Q. Okay. Now, as I understand it, all of
12:51:58 the plug-ins basically come with the basic
12:52:01 Juniper system, like the 5800 SRX box; right?
12:52:05 They're built in?

12:52:06 A. The -- the SRX 5800 basically has, you know, a monolithic static code image. You pretty much get everything. There are no differences in images.

Q. Got it.

A. There's only one image.

Q. Okay. And when, through configuration, different plug-ins are enabled, the system selects which plug-ins to pick or which plug-ins to omit?

MR. KAGAN: Objection. Vague. Incomplete hypothetical.

THE WITNESS: You know, I wouldn't describe it in that way. All of the plug-ins are in memory, they're part of the static code image that we just talked about.

So the question is simply which sequence traffic for a particular -- that meets a particular five tuple in the policy, in what sequence it basically traverses those plug-ins, whether -- again, whether they be kind of the explicit plug-ins that you think of or the ancillary plug-ins that I mentioned earlier.

BY MR. HOSIE:

Q. Okay. Is there a portion of the box

12:55:51 admin is going to say, "Enable protocol parser."
12:55:54 Q. Right. Because they don't know what
12:55:56 that means.
12:55:57 A. They don't know what the hell that
12:55:58 means. And they don't know what -- what
12:55:58 plug-ins would require that.
12:55:59 So, as a result, the programmer, the
12:56:04 developers of the system, basically look at the
12:56:07 combinations. If I have IPS with ALGs, does IPS
12:56:12 come before ALG or does it come after ALG? If I
12:56:15 have IPS and NAT, which one comes first?
12:56:17 So all of these com- -- combinations
12:56:19 have been thought out in advance.
12:56:21 Q. So there's a logical order, a sequence?
12:56:23 A. There's a logical sequence that these
12:56:25 things need to run in. It isn't enough to
12:56:27 simply say that these things are, quote,
12:56:29 enabled --
12:56:29 Q. Right.
12:56:30 A. -- and as a result of it, you know, we
12:56:32 can shop -- shop the packet around willy-nilly.
12:56:34 Q. Right. Because certain things have to
12:56:36 go before other -- other things. There's an
12:56:38 order, a logical order.
12:56:39 A. But that order, again, you know, in our

13:00:26 you mentioned, he's pushed commit. At that
13:00:28 point, somewhere in memory, there's a list of
13:00:31 all of these policies kept --
13:00:33 A. Correct.
13:00:34 Q. -- correct?
13:00:34 And at that point, the system looks at
13:00:36 all of these policies in this service chain --
13:00:39 A. Looks at each policy.
13:00:41 Q. In the service chain --
13:00:42 A. Not in the service chain.
13:00:44 Q. Okay.
13:00:45 A. There's a list of -- there's a policy.
13:00:47 That policy applies to anything that goes from
13:00:49 security zone A to security zone B.
13:00:51 Q. Okay.
13:00:52 A. Within that policy, there are rules.
13:00:53 Q. Okay.
13:00:54 A. Those rules have five tuples in them.
13:00:56 Q. Okay.
13:00:56 A. For each rule, there's a set of actions
13:00:59 that you might take.
13:01:00 Q. Okay.
13:01:00 A. You're going to go ahead and precompute,
13:01:04 effectively -- well, not precompute. You're
13:01:06 going to go select for that -- if that rule gets

13:01:11 triggered and these features are requested and
13:01:14 to your point in this example that you're
13:01:16 giving, all features have been selected. Right?
13:01:18 Q. Uh-huh.
13:01:19 A. You're going to basically say, at that
13:01:21 point, I'm going to select that service chain,
13:01:23 that -- that is basically just my static service
13:01:25 chain that I'm always going to run all flows
13:01:28 through that meet the criteria of this packet.
13:01:31 Right?
13:01:31 Q. Right.
13:01:32 A. So when the packet -- when that packet
13:01:33 appears in the flow table, I'm going to look in
13:01:36 the policy, when that rule triggers, I'm going
13:01:38 to have this preselected path through the system
13:01:42 that that flow will take.
13:01:44 Q. Okay. And the moment before the first
13:01:48 packet hits the system, what exists in the Junos
13:01:53 box?
13:01:54 A. The policy, the rule, and the selection
13:01:57 of that precomputed -- I mean, the selection,
13:02:00 basically, of that service chain --
13:02:03 Q. Okay. So is it --
13:02:04 A. -- that will be used. When -- it's
13:02:05 basically saying when a packet arrives, when a

13:02:08 flow arrives, that meets this rule --

Q. Do the following?

A. -- do this processing on it.

Q. Do this processing. Okay.

And so you don't have actual data structures instantiated in memory as part of a flow-specific processing packet?

A. Not -- not at that point.

Q. Not at that point because it's pre-first packet?

A. Yeah.

Q. Okay.

A. There is no flow yet. In fact, given that you can have wild cards in these rules, it makes no sense to have a flow-specific one; right? If you had a wild-carded rule, the sequence -- the service chain you would drive something -- drive a flow through, right, would be the same for all flows --

Q. Sure.

A. -- that met those criteria. So it does not make sense to do that on a flow-by-flow basis.

Q. Okay. Okay. And so, then, the first packet comes in?

13:59:57 THE WITNESS: I don't -- yeah. I'm not
13:59:58 sure I would kind of describe it that way. I
14:00:00 think there is -- there is memory allocated in
14:00:03 the session table for each of the individual
14:00:07 flows. There may be additional things that are
14:00:10 linked off of those tables, but I'm not -- I
14:00:14 know, like, for things like -- IPv6 is an
14:00:16 example, there will be a pointer off to an IPv6
14:00:21 block, and I think that may be more dynamically
14:00:23 allocated.
14:00:23 BY MR. HOSIE:
14:00:23 Q. Once memory is allocated on a
14:00:26 flow-specific basis, then you have a stateful
14:00:28 instantiated data processing path in your
14:00:30 system?
14:00:30 MR. KAGAN: Objection. Vague.
14:00:32 THE WITNESS: That's a mouthful.
14:00:36 When -- when we -- so we allocate memory on an
14:00:42 as-needed basis; right?
14:00:43 BY MR. HOSIE:
14:00:43 Q. Post-first packet.
14:00:45 A. Post -- so post-first packet, I tend to
14:00:49 think of there being -- there seldom being
14:00:51 memory allocation. I think most of the --
14:00:56 most -- so the slot that's allocated to you in

14:09:04 arrives.

Q. At -- where -- where you can look at the first packet and say, "Okay. This is a flow, this is what it needs, let's allocate memory."

A. When I can look up policy. I mean, the point at which I can look up policy. So as we kind of discussed in the TPC case, it's actually probably at the point that the third packet has arrived at the box --

Q. Given a handshake?

A. -- we've done the handshake, now we basically go up and we're going to look up policy.

Q. And then allocate memory according to what you need?

A. And then for anything that is -- that is not statically allocated, basically, I think the plug-ins -- and it's not so much at a central point within the system. I expect that each plug-in would logically make its own determination for anything that it needs -- needs to dynamically maintain. So --

Q. Right.

A. -- it's going to vary, again, from plug-in to plug-in to plug-in.

14:09:44 Q. Okay. But at that point, as the first
14:09:46 packet arrives, when you have a flow in the
14:09:48 system, allocate dynamically what needs to be
14:09:50 allocated dynamically?
14:09:51 MR. KAGAN: Objection. Misstates
14:09:52 testimony. Vague.
14:09:54 THE WITNESS: I think, at that point,
14:09:56 each plug-in will make its own determination,
14:09:58 and it -- and it may be on the first packet it
14:10:02 ever sees, it may be on the tenth packet it
14:10:04 sees, it may be on the hundredth packet it sees.
14:10:07 BY MR. HOSIE:
14:10:07 Q. Do you know the implementation details
14:10:08 of that, sir, how the plug-ins allocate memory
14:10:11 or have memory allocated for them?
14:10:13 A. No. I do not.
14:10:14 MR. HOSIE: Okay. Why don't we take a
14:10:16 break.
14:10:18 THE VIDEOGRAPHER: We're off the record
14:10:20 at 2:10 P.M.
14:10:22 (A brief recess was taken.)
14:15:24 THE VIDEOGRAPHER: We're back on the
14:23:39 record at 2:23 P.M. in the deposition of
14:23:42 Mr. Oliver Tavakoli. Please continue.
BY MR. HOSIE:

I, BRENDA L. MARSHALL, Certified Shorthand Reporter, License No. 6939, do hereby certify:

That, prior to being examined, the witness named in the foregoing deposition, to wit, OLIVER TAVAKOLI, was by me duly sworn to testify the truth, the whole truth and nothing but the truth:

That said transcript was taken down by me in shorthand at the time and place therein named and thereafter reduced to computerized transcription under my direction.

I further certify that I am not interested in the event of the action.

WITNESS this 3rd day of July, 2012.

_____

BRENDA L. MARSHALL

EXHIBIT D
FILED UNDER SEAL

EXHIBIT E

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.,

Plaintiff,

vs. No. C10-4234 SI

JUNIPER NETWORKS, INC.,

Defendant.

CONTAINS CONFIDENTIAL PORTION

VIDEOTAPED DEPOSITION OF SCOTT NETTLES, PH.D.

San Francisco, California
Friday, October 19, 2012
Volume I

Reported by:
SUZANNE F. BOSCHETTI
CSR No. 5111

Job No. 1540467

CONFIDENTIAL PORTION: 165 - 199
PAGES: 1 - 297

configurable at some level or another.

Q Have you seen the book called Junos Security?

A Yes, I have.

Q Do you have a copy of this book? 02:28:55

A I have several copies.

Q Do you have a copy of this book in which you've marked it up with highlighting?

A No, I don't like to highlight books.

(Deposition Exhibit 221 marked by the court reporter.) 02:29:08

BY MR. MCPHIE:

Q I'm handing you what has been marked Exhibit 221, excerpts from a book Junos Security. Do you consider the book Junos Security to be a reliable source? 02:29:44

A Well, it was published by O'Reilly and Juniper Networks themselves. It's widely available. I bought my two copies on Amazon. My understanding is that your expert testified that it's an authoritative source. And it seems to be a book that Juniper has published to inform its customers and its -- their sys admins how to use the SRX Services Gateways. It seems like an extremely reliable book. 02:30:12 02:30:33

Q You consider the Junos Security book to be an extremely reliable text?

A It would appear to be, yes.

Q Turn to the last page of Exhibit 221 and you'll see there's some underlined language there. Do you recall highlighting this language in a version of a Junos Security book? 02:30:48

MR. HOSIE: Objection. Lacks foundation.

THE WITNESS: No, sir, I -- I have -- I have no idea where you got this, this underlined book, but it's not mine. 02:31:09

BY MR. MCPHIE:

Q Could you please read the underlined language, and tell me whether you think it's accurate. 02:31:18

A Can you explain to me why you think this is my book?

Q What's that? I'm just asking questions.

A I just don't understand why you're suggesting that I've underlined something that I know that I didn't. It seems to be a -- a false accusation. 02:31:28

MR. HOSIE: He's not.

MR. MCPHIE: Hold on.

MR. HOSIE: He's just asking a question and 02:31:34

(Reporter's clarification.)

Sorry, I apologize.

"JUNOS Software is a single network operating system integrating routing, switching, and security. Most Juniper Networks hardware platforms run JUNOS Software (herein JUNOS)."

Then it goes on to talk a little bit more about JUNOS. And we know for a fact that in addition to the MultiServices PICs -- I mean, they're part of a router -- in addition to those routers running JUNOS, that the J series routers and the SRX series routers run JUNOS. So, you know, that seems clear that Pavel -- the understanding here was that this was about how JUNOS worked. 03:12:50

Q And, in fact, it was clear in your mind upon carefully reviewing Exhibit 222 that the analysis, the detailed analysis of Exhibit 222 applied to each and every one of the Juniper accused products, right? 03:13:17

MR. HOSIE: If I could have that read back, please.

MR. MCPHIE: I can read it.

BY MR. MCPHIE:

Q And, in fact, it was clear in your mind 03:13:32

I, the undersigned, a Certified Shorthand Reporter of the State of California, do hereby certify:

That the foregoing proceedings were taken before me at the time and place herein set forth; that any witnesses in the foregoing proceedings, prior to testifying, were duly sworn; that a record of the proceedings was made by me using machine shorthand which was thereafter transcribed under my direction; that the foregoing transcript is a true record of the testimony given.

I further, certify I am neither financially interested in the action nor a relative or employee of any attorney or party to this action.

IN WITNESS WHEREOF, I have this date subscribed my name.

Dated: 10/26/12

_____

SUZANNE F. BOSCHETTI

CSR No. 5111

EXHIBIT F

MILLER & COMPANY REPORTERS

CERTIFIED TRANSCRIPT
CONFIDENTIAL

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

IMPLICIT NETWORKS, INC., Plaintiff, vs. JUNIPER NETWORKS, INC., Defendant.	No. C 10-4234 SI

CONFIDENTIAL TRANSCRIPT

DEPOSITION OF: PETER ALEXANDER, Ph.D.

TAKEN ON: October 16, 2012

NO. 13235

REPORTED BY: BEVERLY L. NEWMAN
CSR No. 2872

being from that document.

Q All right, sir. So is Juniper accurately describing the way its systems work in this graphic it put in its brochure to its customers?

MR. McPHIE: Objection. Vague and ambiguous.

THE WITNESS: Oh, I think it's accurate in terms of the purpose of this brochure, which is a marketing view of the Juniper products.

So if you're talking about accuracy, you know, I prefer to think in terms of accuracy in scientific terms. But this is a marketing document, so it's probably accurate in that sense.

BY MR. HOSIE:

Q And you see at the bottom it says, "One OS, One Release Track, One Architecture"? Do you see that?

A Yes, I do.

Q True statements; right?

MR. McPHIE: Objection. Vague and ambiguous. Compound.

THE WITNESS: Well, I know for sure that I think it's this document that refers to one code base, and so if they are talking about one OS being one code base from which products are created by selecting various elements of the code base, it's accurate.

///

REPORTER'S CERTIFICATE

I, BEVERLY L. NEWMAN, CSR No. 2872, certify:

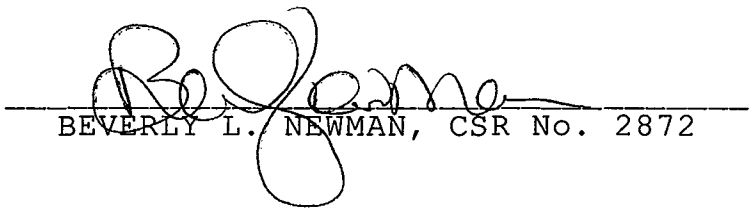
That the foregoing deposition of PETER ALEXANDER, Ph.D. was taken before me at the time and place therein set forth, at which time the witness was put under oath by me;

That the testimony of the witness and all objections made at the time of the deposition were recorded stenographically by me and were thereafter reduced to a computerized transcript under my direction;

That the foregoing transcript is a true record of the testimony of the witness and of all objections and colloquy made at the time of the deposition.

I further certify that I am neither counsel for nor related to any party to said action nor in anywise interested in the outcome thereof.

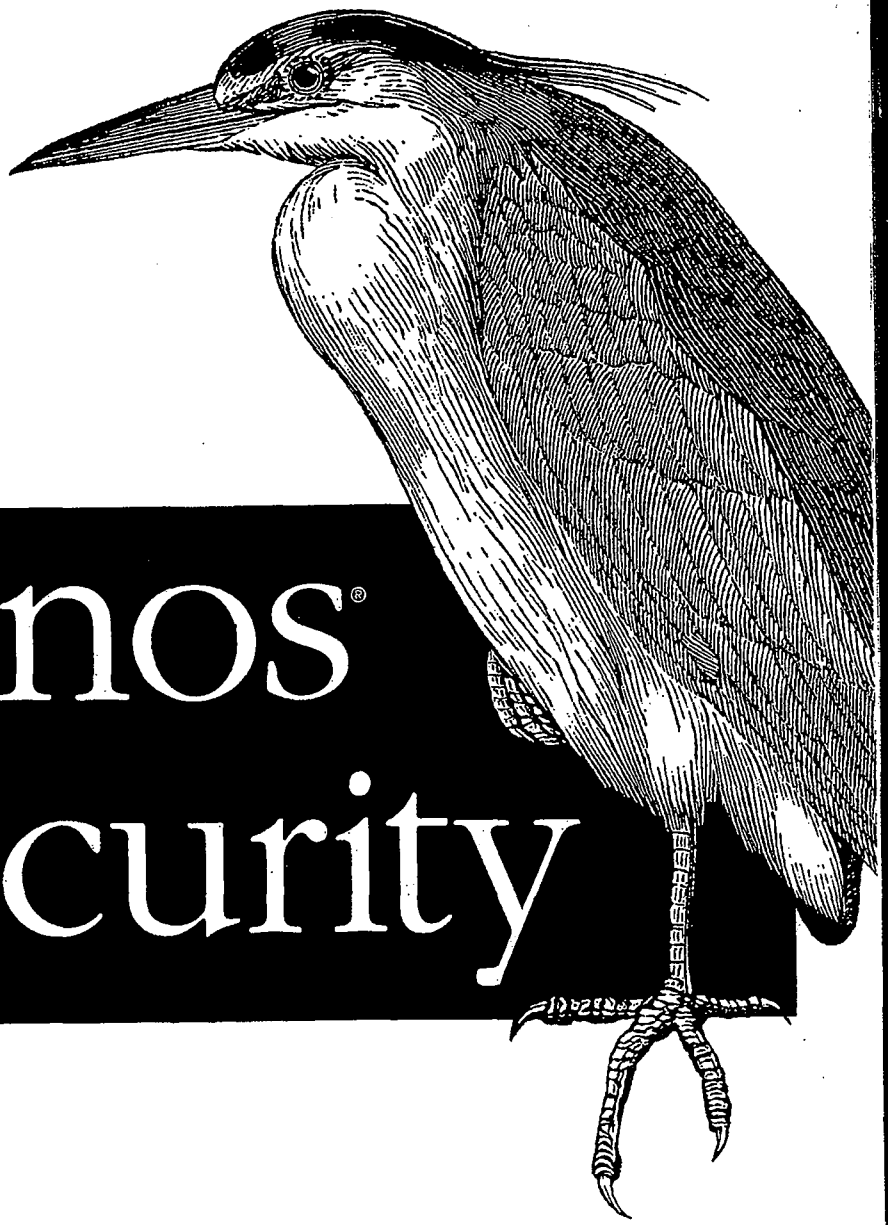
IN WITNESS WHEREOF, I have subscribed my name this 19th day of October, 2012.



BEVERLY L. NEWMAN, CSR No. 2872

EXHIBIT G

A Guide to Junos for the SRX Services Gateways & Security Certification



Junos® Security

Rob Cameron,
Brad Woodberg, Patricio Giecco,
Tim Eberhard & James Quinn

O'REILLY® JUNIPER NETWORKS

Preface

Juniper Networks built the SRX Series as an answer to the network and security challenges of today that would be ready to scale and adapt to the inevitably larger and more complex demands of tomorrow. Security remains a huge and still growing challenge for any organization grappling with modern communication networks. Whether it is the explosion in traffic (good *and* bad), the growing complexity of data centers and cloud computing, or the menacing evolution of threats to that infrastructure, the days of the simple firewall are over. Something radically new was needed, and the SRX is leading the charge into a more secure future.

Junos Security is your guide to this brighter future. It readily answers the questions you have, will have, or may even hope to have. The SRX is one awesome beast that is up to matching your challenges whether they are firewalling, routing, NAT, deep inspection, encryption, or the mitigation of nearly any form of network attack.

How do you write about such a thing? Once upon a time, there were firewall books, or routing books, or even data center deployment books. But today, this one book is here to illuminate the elaborate hybrid workings of this next-gen networking marvel. Add to that the fact that the SRX platform has multiple models across two quite distinct device classes covering everything from the smallest networks in the world to the very largest, along with the huge and legendary heritage of the Junos operating system, and you have more than enough material to fill many volumes of books.

Writing a book of this magnitude was no easy task to undertake. In fact, it took five of the best SRX engineers in the world to accomplish it, collaborating for almost a year. Together they have many times more man-years of experience working with the SRX than the device has even existed, so they bring a real-world approach in this book that you can take away to your own work immediately.

Ultimately, this book is about Junos and the SRX, and how to deploy, configure, and maintain your Juniper Networks investment with the goal of protecting and efficiently operating your network. Enjoy!

Firewall and security concepts

A high-level understanding of firewall and security concepts is helpful. We will go into detail about best practices and how these can be implemented on the SRX.

Routing

This includes basic knowledge of routing protocols and dynamic routing principles.

Point-to-point links

These network segments are often thought of as WAN links in that they do not contain any end users. Often these links are used to connect routers together in disparate geographical areas. Possible encapsulations used on these links include ATM, Frame Relay, PPP, and HDLC.

IP addressing and subnetting

Hosts using IP to communicate with each other use 32-bit addresses. Humans often use a dotted decimal format to represent this address. This address notation includes a network portion and a host portion which is normally displayed as 192.168.1.1/24.

TCP and UDP

These Layer 4 protocols define methods for communicating between hosts. TCP provides for connection-oriented communications while UDP uses a connectionless paradigm. Other benefits of using TCP include flow control, windowing/buffering, and explicit acknowledgments.

ICMP

This protocol is used by network engineers to troubleshoot and operate networks as it is the core protocol used by the ping and traceroute (on some platforms) programs. In addition, ICMP is used to signal error and other messages between hosts in an IP-based network.

P2.2. What's In This Book?

This book was written to be the definitive and most complete source of information for working with the SRX platforms. It is divided into 13 chapters. Each chapter is written by one of the authors from our authoring pool of five. While we tried to review each other's work, you'll be able to tell different voices in the writing styles, and we hope that this is generally refreshing rather than a hindrance.

Here is a detailed accounting of what's in this book:

Chapter 1

The SRX is Juniper Networks' next-generation services platform. The devices combine the advanced Junos operating system with the existing security offerings on a high-speed feature-rich platform. This chapter is designed to give you an understanding of the physical devices as well as their architecture. Then it walks you through common deployment scenarios and use cases. The enriching explanation provides a clear vision into the platforms and strategies that are available when using the SRX platforms.

Chapter 2

Junos is one of the industry's most well-respected network operating systems. Over its 10-plus-year history, Junos has grown into a feature-rich platform. Because Junos and its capabilities are so large, it's important to build a strong base of knowledge of what Junos is all about. In this chapter, the design of the Junos operating system, its fundamental concepts, and its history are discussed. Also, for readers who are coming from other platforms, a comparison between other major firewall platforms is drawn to Junos on the SRX.

Chapter 3

Using Junos requires the use of hands on a keyboard. This chapter gets you hands-on with

EXHIBIT H

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.

Plaintiff,

v. Case No. C 10-4234 SI

JUNIPER NETWORKS, INC.

Defendant.

HIGHLY CONFIDENTIAL - UNDER PROTECTIVE ORDER

VIDEOTAPED DEPOSITION OF SCOTT M. NETTLES, Ph.D.

San Francisco, California

October 9, 2012

Reported by:

KENNETH T. BRILL

CSR NO. 12797

Job No. 1538661

PAGES 1 - 285

relied upon are final, and I've also reviewed the exhibits thereto, so -- and that's a -- that's a design document involving I think the right word is -- is Viking. 11:15:30 11:15:33 11:15:37 11:15:40

Q. Mm-hmm. Are there any -- well, withdrawn. 11:15:41

What aspects of the Krishna N. deposition exhibits do you believe support your opinions regarding infringement but were not cited in the report? 11:15:44 11:15:53 11:15:56 11:16:00

A. Well, in general, that document is a design document about what eventually became the SRX series of products, which are some of the main products that were accused. And there are numerous diagrams in that -- in that document that would support my opinions further. 11:16:03 11:16:07 11:16:12 11:16:14 11:16:17 11:16:20

The -- a specific table that I was -- had in mind was there's an enumeration of a series of application level gateways and a discussion of the amount of state that they would allocate on a per-session basis. 11:16:24 11:16:28 11:16:33 11:16:42 11:16:47

Q. Was there a number or letter attached to this table that you can recall? 11:16:50 11:16:53

A. I -- I would have to -- I would have to look at my -- at my copy, or if you have a copy, I'd be glad to look at your copy. 11:16:56 11:17:00 11:17:03

in a different section of the report to support element 1g of Claim 1 of the '163 patent for the SSL component? 04:08:48 04:08:53 04:09:01

MR. HOSIE: Objection, asked and answered. 04:09:02

THE WITNESS: I already answered that. I mean, the place that I was reading from before, I've lost that place now, talks about SSL in the particular context it's talking about it doing decryption. You can't do decryption without reading and writing and manipulating state. 04:09:03 04:09:05 04:09:07 04:09:11 04:09:14 04:09:17

BY MR. McPHIE: 04:09:19

Q. But you don't state that in that section of the report, do you? 04:09:22 04:09:25

A. I don't think that there is any place in my report -- and I'm glad to look and probably I should, where I say anybody who knows anything about decryption which SSL does, knows that that's going to require reading and writing state. You know, there's -- there's a lot of disclosure about this. I don't have to lead your expert by the -- by the nose and say, look at this thing that obviously reads and writes state. It obviously reads and writes state. 04:09:37 04:09:38 04:09:40 04:09:44 04:09:48 04:09:52 04:09:55 04:09:59 04:10:02 04:10:06

I understand you would have been happier if I had done that but, you know, I don't think I 04:10:06 04:10:08

report for the evidence of this. On page 41, about two-thirds of the way down the page, there is a discussion which says, Secure sockets labl- -- layer SSL is a cryptographic protocol that adds security to TCP/IP communication. Several versions of SSL and transport layer security TLS protocols are in widespread use in applications like web browsing, electronic mail, Internet faxing, instant messaging and voice over IP, VOIP. 04:25:07 04:25:20 04:25:23 04:25:25 04:25:30 04:25:35 04:25:40 04:25:43 04:25:47

SSL and TLS encrypt the transport layer protocol diagrams that carry the payload of these communications. While encryption is an excellent way to keep private data from prying eyes, without inspection by the IDP series device, it also unwittingly opens the network to dangerous viruses, trojans, or network attacks. To inspect the HTTP payload of the HTTPS traffic, the IDP series device must decrypt the HTTPS session. Your security policy can examine both the SSL session and the decrypted HTTP payload. 04:25:51 04:25:55 04:25:58 04:26:01 04:26:04 04:26:08 04:26:08 04:26:15 04:26:21 04:26:26 04:26:29

So this is an example of SSL manipulating state as required by the last limitation of Claim 1. 04:26:32 04:26:35

Q. Now, this is in Section 1b; correct? 04:26:43

A. And I believe that what I said was throughout the report. 04:26:46 04:26:48

referring to. 05:02:44

BY MR. McPHIE: 05:02:53

Q. Can you identify the piece of supporting evidence that you are looking to reading that evidence in its entirety, please. 05:02:54 05:02:55 05:03:05

A. Well, on page 8, I would point to the figure that's at the top. 05:03:13 05:03:15

Q. And specifically, what aspect of that figure indicates to you that state information stored for one pocket is then used in processing a subsequent packet? 05:03:17 05:03:18 05:03:26 05:03:29

A. Well, I think the entire fast path. 05:03:48

Q. And what is the component associated with that state information? 05:03:51 05:03:55

A. Well, each of the components that make up the fast path. 05:04:00 05:04:02

Q. Which in this case was what? 05:04:09

A. Well, I mean, there's a lot of different components here. There's screens, there's TCP, there's NAT. Those might actually also be composed of subcomponents. There's services. There's ALG. Those are definitely composed of subcomponents, but I think that -- I don't know if all of the subcomponents necessarily do the stateful requirements, but many of them do, and certainly 05:04:10 05:04:13 05:04:18 05:04:23 05:04:30 05:04:36 05:04:41 05:04:44

the -- the other ones do. 05:04:47

Q. What is it about this diagram that 05:05:03
suggests to you, for example, for the NAT component, 05:05:05
that it stores state information that is used for 05:05:10
processing a subsequent packet? 05:05:13

A. Well, you didn't ask me that question 05:05:25
before, but I read about how NAT works. The most 05:05:27
obvious place would be in Junos Security, but 05:05:30
probably in a number of other -- probably in a 05:05:34
number of other of the documents that are cited, and 05:05:36
I know that in junos-nat and actually, almost as far 05:05:39
as I can tell, any module that's sort of this level, 05:05:43
can do logging. And so logging would be an example 05:05:49
of -- of that for NAT. 05:05:52

Q. Do you point to NAT logging at any point 05:05:56
in your report? 05:06:00

A. No, not explicitly that I remember, but 05:06:02
I'd be glad to look if you'd like me to. 05:06:05

Q. What I'm looking for is a -- a specific 05:06:08
example of a piece of state information cited in 05:06:13
your report that, in fact, is stored and then used 05:06:22
for a subsequent packet. Could you identify one 05:06:24
such piece of evidence by page or paragraph number 05:06:28
only? 05:06:32

A. Well, I think I just did that, but I'll be 05:06:39

CERTIFICATE OF REPORTER

I, KENNETH T. BRILL, a Certified Shorthand Reporter, hereby certify that the witness in the foregoing deposition was by me duly sworn to tell the truth, the whole truth, and nothing but the truth in the within-entitled cause;

That said deposition was taken down in shorthand by me, a disinterested person, at the time and place therein stated, and that the testimony of the said witness was thereafter reduced to typewriting, by computer, under my direction and supervision;

I further certify that I am not of counsel or attorney for either or any of the parties to the said deposition, nor in any way interested in the event of this cause, and that I am not related to any of the parties hereto.

DATED: 10/24/2012

_____

KENNETH T. BRILL

CSR#12797

EXHIBIT I

UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

IMPLICIT NETWORKS, INC.,	)	
	)	
Plaintiff,	)	
	)	
vs.	)	No. C 10-4234 SI
	)	
JUNIPER NETWORKS, INC.,	)	
	)	
Defendant.	)	
	)	

HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY

30(b)(6) DEPOSITION OF: OLIVER TAVAKOLI

TAKEN ON: June 19, 2012

VOLUME I: Pages 1 through 128, inclusive

13145

BRENDA L. MARSHALL

CSR No. 6939

15:03:38 group inside of SBU --

15:03:42 Q. Okay.

15:03:42 A. -- which is where the customer-specific testing would take place.

15:03:45 Q. Okay. You said you're familiar with some of the configuration for some of the

15:03:49 customers. Which ones are you familiar with?

15:03:51 A. Charles Schwab, Morgan Stanley, Verizon, AT&T, Vodafone, 7-Eleven, Payless Shoes. You

15:04:00 want me to keep going?

15:04:15 Q. Sure.

15:04:16 A. UniCredit, Orica --

15:04:17 Q. Spell it, please.

15:04:35 A. -- O-r-i-c-a -- Motorola.

15:04:36 Those are the ones that kind of come off the top of my mind in the last, you know, month,

15:04:45 month and a half.

15:04:46 Q. Fair enough. Schwab, what Juniper boxes has Schwab purchased, please?

15:04:48 A. Schwab has deployed primarily SRX 5Ks. Some of them are 3Ks. We deploy them primarily

15:04:53 in cluster mode at the data center perimeter.

15:04:55 Redacted

15:05:00

15:05:03

15:05:10

15:05:12

15:05:15

15:08:42 Q. Okay.

15:08:42 A. -- for our designs.

15:08:43 Q. All right. And before Schwab brought the 5 carries -- 5K series boxes online, did it provide Juniper with a series of configuration requests so that Juniper could test the boxes?

15:09:00 A. [redacted]

They have resident engineers on site, and they have their advanced TAC contact.

Q. What do you mean -- what do you mean, resident engineers on site?

A. These are engineers that are Juniper-badged employee -- they're actually a dual-badged employee. They're paid for by Juniper.

Q. Yes.

A. They're typically dual-badged. So these would be -- they would carry a Verizon, kind of, contractor badge and a -- and a Juniper badge, and they would basically be usually paid for by the customer, although, again, if you buy enough stuff, you may get them for free, and Juniper might eat the cost of the resident engineer.

And, ultimately, the goal for these resident engineers is to be Juniper's eyes and ears on the ground. And the value to somebody like Verizon is that they get a clear conduit back to Juniper that has, like, no static on it, that can translate exactly what the problem is that they have with, you know, a solution that they need.

Q. Fair -- fair enough. How many resident